

From Packet to Power Switching: Digital Direct Load Scheduling

Mahnoosh Alizadeh*, *Student Member, IEEE*, Anna Scaglione*, *Fellow, IEEE*
and Robert J. Thomas†, *Life Fellow, IEEE*

Abstract—At present, the power grid has tight control over its dispatchable generation capacity but a very coarse control on the demand. Energy consumers are shielded from making price-aware decisions, which degrades the efficiency of the market. This state of affairs tends to favor fossil fuel generation over renewable sources. Because of the technological difficulties of storing electric energy, the quest for mechanisms that would make the demand for electricity controllable on a day-to-day basis is gaining prominence. The goal of this paper is to provide one such mechanisms, which we call Digital Direct Load Scheduling (DDLS). DDLS is a direct load control mechanism in which we unbundle individual requests for energy and digitize them so that they can be automatically scheduled in a cellular architecture. Specifically, rather than storing energy or interrupting the job of appliances, we choose to hold requests for energy in queues and optimize the service time of individual appliances belonging to a broad class which we refer to as “deferrable loads”. The function of each neighborhood scheduler is to optimize the time at which these appliances start to function. This process is intended to shape the aggregate load profile of the neighborhood so as to optimize an objective function which incorporates the spot price of energy, and also allows distributed energy resources to supply part of the generation dynamically.

Index Terms—Demand side management, aggregator, load scheduling, electric vehicles, Smart Grid communications

I. INTRODUCTION

The most important rule of power systems is that demand and generation should be balanced at all times. In order to fit into the secure operational framework of today’s power grid, generators need to be able to forecast and control their output power for the next day and commit to generating the amount of electricity that they are scheduled to produce by the independent system operator (ISO). Failure to do so will result in system reliability issues; therefore, to ensure compliance, stiff penalties are imposed on those not able to meet their schedule. This favors the predictable and controllable power which can be generated from fossil fuels. With today’s mostly price-inelastic demand, incorporating considerable amounts of renewable generation, given the volatility and unpredictability of such sources, is a challenge [1].

One way to overcome the dispatch issues associated with renewable resources is through a cyber system in support of making energy demand both observable and controllable. One embodiment of the so-called Smart Grid is a model for *smart* devices to become flexible agents, distributed across the grid, whose demand can be modified, either in a centralized or a distributed manner, via programs that produce market efficiency.

These flexible agents could, among other things, modulate the demand to more easily integrate volatile renewable generation sources in the system, especially those available locally, at the edge of the distribution network. To help this remarkable transition, utility companies are introducing smart meters, whose deployment is referred to as the Advanced Metering Infrastructure (AMI) [2]). The AMIs are currently used to improve the prediction accuracy of the load and curtail demand during emergencies. The question is *when, how and with what incentives* load control could be performed on a regular basis.

A. Previous Work

Previous work on demand management techniques can be categorized into either price-based load control techniques, which fall under the class of Demand Response (DR) methods, or direct load control (typically through curtailment), which is classified as Demand Side Management (DSM).

Price-based load control strategies include Time Of Use (TOU), Critical Peak Pricing (CPP), or Real-Time Pricing (RTP) techniques, all of which require customers to make energy usage decisions individually based on pricing signals. In TOU pricing strategies, the price data is usually delivered months or years before the actual time of use. There have been several studies on determining these rates, which requires a dynamic model for the price response of customers, typically derived based on experiments [3], [4]. TOU can ameliorate customer behavior and smoothen the daily demand profile, but has little chance of aiding the integration of volatile resources. In RTP, instead, price information is provided only hours before consumption. With RTP, there is a need for an automated system to help the customer make energy usage decisions; generally they are referred to as Home Energy Management Systems (HEMS) (see e.g., [5], [6]). HEMS plan the customers consumption given pricing data, appliance power profiles, job deadlines, and user preferences. RTP strategies are appealing since they correct inefficiencies with a decentralized market driven control, but raise some concerns in terms of safety. Real-time prices are not actually real-time since they have to be delivered to the customer beforehand, to allow some planning time. Also, the presence of a feedback loop due to customer response may result in rebound peaks that can worsen the situation [7] and can also lead to physical and price instability [8]. An important question is how to compute the optimum price for each end-use customer without introducing economic and physical safety issues. Can these prices be equal for every customer, or type of appliance, in a neighborhood, or should they change? Should they be updated continuously, or in an event-driven fashion? Determining pricing also requires extensive knowledge about customer behavior: on pre-determined rates, customers are much more predictable.

*Department of Electrical and Computer Engineering, University of California Davis, email: {malizadeh,ascaglione}@ucdavis.edu, †Department of Electrical Engineering, Cornell University, email: rjt1@cornell.edu. This work has been funded by DOE under CERTS. Parts of this work was presented at Smartgridcomm 2011.

Demand Side Management strategies are, in contrast, built for safety. They are usually applied directly by a control center, and require customer subscription to an economic incentive program. The first techniques emerged in the 60's and are currently employed through the so-called *Interruptible Load* programs, where some of the customers appliances, upon receiving a notice, are automatically turned off for a pre-determined amount of time (15-30 minutes). Several methods have been proposed to determine the best load curtailment strategy [9], [10]. These schemes have the benefit of aggregating several load assets but their pitfall is that it is hard to determine off-line the degree to which they inconvenience the customers. Thus, these programs are normally used only in emergency situations.

In recent years, the deployment of advanced two-way communication links are leading to rethink *centralized load control*. Rather than the utility, an aggregator could directly control end-use appliances and tap not only customers HVAC (heating, ventilation, and air conditioning) systems but also on their electric vehicle (EV) batteries [11]. Next, we explain that EVs and HVAC have significant differences and, in our view, require separate treatment.

B. Classification of Loads and research on Deferrable Loads

There are three main classes of appliances used by residential customers. A first class of automata is one whose load profile, once on, is predictable. This class includes, for example, EVs, dishwashers, washers/dryers etc. Given that their cycles last for a long time, their starting time can be easily shifted, but their operation should not be interrupted. In this paper we specifically target this first class of *deferrable loads*. A second class of loads has a predictable profile except for an unknown duration. This class includes lighting, television sets, stereos etc. The service these loads deliver is extremely time sensitive, and so is their electricity demand. Therefore, the demand due to this type is not very flexible (except for dimmable lights). Last but not least, the third class of loads includes, for example, thermostatically controlled appliances. These loads can be both interrupted and turned on earlier, but their load profile and the time sensitivity of the service depend non-linearly on a local state. Most centralized load control programs interrupt this class of appliances. Here we argue that there is an additional layer of complexity in capturing digitally their service model, and therefore, to avoid incorrect generalization, we currently exclude them from our treatment.

Specifically, we will name as *D-loads* those deferrable loads which, if enabled by embedded logic and two way communications, can specify their energy request to a remote control center and wait to be dispatched.

The most important *D-loads* are Electric Vehicles (EV). Researchers have proposed algorithms that control the energy use of EVs under dynamic pricing [12], [13]. Examples of direct control are the work in [14], where the charging rate of EVs drawing from a single feeder is optimized by dividing them into service queues depending on their desired time of departure; the work in [15], instead, optimizes the charging rate of EVs by including them in the optimal power flow problem solved by the system operator. The inclusion of deferrable

loads in the energy market in the presence of renewables is the subject of [16], [17]. In this line of work, the demand response assets are modeled as an aggregate tank/storage capacity that needs to be filled by a certain deadline. The DSM model we propose in this paper can provide a more accurate description of the load modification capability of an aggregator, and for the information flows to support the service.

C. Contribution

In this paper, we propose a direct load management scheme that mediates between the central control model of DLC and the *laissez faire* nature of RTP. The scheme we propose in this paper is a direct control architecture that can be implemented as a voluntary program in which customers release the control of the time at which their D-loads start functioning to their associated neighborhood controller or *aggregator*. They turn over this control in return for a financial reward for the inconvenience they experience, as their energy use is manipulated to follow a desired demand profile closely. We envision that the scheduling decisions are made at several distributed control centers that aggregate energy requests from many customers and make energy usage decisions that are relayed back to them. Anticipating that aggregating energy requests up to a certain level will give us the benefits of scale, our model allows us to explore to what degree cooperation among customers benefits the efficiency of the system. In fact, our scheme can be viewed as a model for building up *reservoirs of D-loads requests*, like a virtual water dam for loads, which can be dispatched optimally to follow the generation, thereby reversing the traditional roles played by generators and loads.

Our contribution is the introduction of a Digital Direct Load Scheduling (DDLs) model to manage D-loads and the network architecture that incorporates it. The model includes 1) an *encoding component* for controllable load, generation and for the feedback signals, 2) a *load scheduler* and 3) a *generation market interface*. The encoding component allows deferrable appliances to become D-loads, which means that through the DDLs these appliances become capable of specifying their request for energy and release the control of their starting time to the scheduler, which determines their activation time in such a way to shape the load profile to follow available green generation. In this paper, we present the encoding component and the load scheduler aspects of our proposed model. The proposed feedback mechanism informs the D-loads that need to be activated about the time they can start their job. The description of how the proposed management system interacts with the electricity market and the extension of the DDLs to thermostatically controlled appliances is left as future research.

II. THE LOAD MODEL

The load in the power grid is a mixture of random requests. In our model we separate the D-loads portion from the remaining part of the load. In fact, the load is:

$$L(t) = L^N(t) + L^S(t), \quad (1)$$

where $L^N(t)$ represents the base load, which we presumably have no control over, while $L^S(t)$ is the *controllable part of the*

load due to the D-loads. Here, we consider $L^N(t)$ predictable using standard load forecasting methods (e.g., [18], [19], [20]).

The next step is to further fragment $L^S(t)$ into contributions from individual appliances. In the model, D-loads arrive in the system randomly following a non-stationary arrival process:

$$a(t) = \sum_{i=1}^{\infty} u(t - t_i^a), \quad (2)$$

with $u(t)$ the unit step, t_i^a the arrival time of the i^{th} D-load of any type, and an arrival rate of $\lambda(t)$ for these appliances. Each arrival event i has an associated parameter vector C_i that determines uniquely the time evolution of the load contribution when that appliance is turned on. In fact, we assume that, if turned on at time zero, the D-load injection is the complex phasor signal $g(t; C_i)$, one to one with C_i . In general, $C_i \in \mathbb{C}^N$ can be the Nyquist samples, or the Fourier or Wavelet coefficients of the known load evolution after activation. A simple example is that of EVs, for which C_i is a two-dimensional vector, representing the charging rate and the fraction of battery capacity needed by the car upon its arrival. C_i 's are assumed to be i.i.d. random vectors with a known stationary probability distribution $f_C(c)$, independent of the arrival times. Thus, the i^{th} arrival event is modeled by a tuple (t_i^a, C_i) . If every arriving D-load is allowed in the system without any delay, the unscheduled demand profile would be:

$$L^{US}(t) = \sum_{i=1}^{\infty} g(t - t_i^a; C_i). \quad (3)$$

The degrees of freedom we will use in shaping the load is to delay the switch-on time for each D-load. Hence, the process of D-loads switching on is a *departure process*, in which the departure time $t_i^d > t_i^a$ is the time instant when the i^{th} appliance is scheduled to switch on:

$$d(t) = \sum_{i=1}^{\infty} u(t - t_i^d), \quad t_i^d \geq t_i^a. \quad (4)$$

In addition, it is convenient to include in $L^N(t)$ the load due to previously scheduled D-loads, since scheduled tasks are non-preemptable and we cannot reschedule them.¹

Thus, the future values of $L^S(t)$ can be written as the following function of future departures

$$L^S(t) = \sum_{i \in I} g(t - t_i^d; C_i), \quad t_i^d \geq t_i^a, \quad (5)$$

where the set I includes all D-loads that have yet not been scheduled, including those that have already arrived in the system at time t but have not yet been authorized to function or the future arrivals.

III. THE ENCODING COMPONENT OF THE DDLS

To enable the optimal scheduling, the i^{th} D-load communicates the tuple (t_i^a, C_i) to the scheduler, upon its arrival. Naturally, this information has to be digitized.

¹This assumption is not as restrictive as it seems, since one could potentially impose an upper-bound to the request made, effectively breaking down the transaction in parts. We believe that this option would be much more practical than assuming jobs can be interrupted arbitrarily.

First, the charging codes C_i are quantized through a mapping $\Psi(C_i)$ onto Q codes C_q , $q = 1, \dots, Q$, forming the codebook $\mathcal{C} = \{C_1, \dots, C_Q\}$. The effect of this mapping is twofold: 1) it obviously provides a digital representation of the incoming request that can be communicated; 2) it separates the D-loads in Q classes of service (queues), one per quantization code. The arrival rate for the q -th queue is:

$$\lambda_q(t) = \lambda(t) \int_{x \in \Psi^{-1}(C_q)} f_C(x) dx, \quad q = 1, \dots, Q.$$

Since arrivals in different queues are independent processes, the arrival process $a(t)$ can be divided into Q separate processes whose state can be represented by a vector $\bar{a}(t) = [a_1(t) \dots a_Q(t)]^T$ of length Q with the property that $\|\bar{a}(t)\|_1 = a(t)$. Each element of this vector is given by,

$$a_q(t) = \sum_{i=1}^{\infty} \delta(\Psi(C_i) - C_q) u(t - t_i^a), \quad (6)$$

where $\delta(\cdot)$ here represents the Kronecker delta function. The corresponding departure processes can also be represented by a vector $\bar{d}(t) = [d_1(t) \dots d_Q(t)]^T$, also satisfying $\|\bar{d}(t)\|_1 = d(t)$. Each $d_q(t)$ is:

$$d_q(t) = \sum_{i=1}^{\infty} \delta(\Psi(C_i) - C_q) u(t - t_i^d). \quad (7)$$

We also know that $\bar{a}(t) \succeq \bar{d}(t)$, where \succeq represents element by element inequality. This is simply due to the fact that the number of departures from each queue can never be larger than the number of arrivals. The quantization of the charge codes allows a simple system representation of the relationship between the individual queue departure processes and the total D-load, that is equivalent to (5) for $Q \rightarrow \infty$. In fact, since the C_q 's are discrete values, the synthesis formula for the quantized load reconstruction is:

$$\hat{L}^S(t) = \sum_{q=1}^Q \sum_{i \in I} \delta(\Psi(C_i) - C_q) g(t - t_i^d; C_q) \quad (8)$$

$$= \sum_{q=1}^Q \frac{\partial}{\partial t} d_q(t) \star g(t; C_q), \quad (9)$$

where $\frac{\partial}{\partial t} d_q(t)$ will produce a Kronecker delta at each t_i^d in the q -th queue and \star is the convolution sign.

In order to work online and compute feasible schedules, the second step is to operate in discrete time intervals. Thus, we assume that the departure process $\bar{d}(t)$ can only have increments at discrete times:

$$t_i^d \in \{\ell \Delta \mid \ell \in \mathbb{N}\}. \quad (10)$$

Given the fact that both t_i^d 's and C_q 's are chosen from a discrete set, we can alternately represent the future D-loads contribution in (5) for $t \geq \ell_0 \Delta = \lceil \frac{t}{\Delta} \rceil \Delta$ as

$$\hat{L}^S(t) = \sum_{q=1}^Q \sum_{\ell=\ell_0}^{\infty} [d_q(\ell \Delta) - d_q((\ell-1)\Delta)] g(t - \ell \Delta; C_q). \quad (11)$$

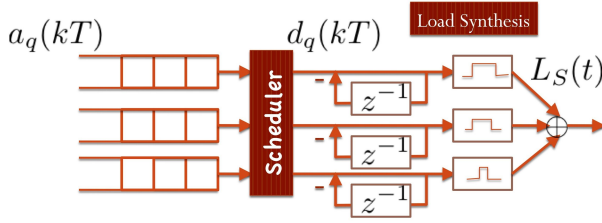


Fig. 1. Diagram of the Load Synthesis Equation

One can further assume that Δ^{-1} is higher than the Nyquist rate needed to sample the charge profile $g(t; C)$ for all possible $\Psi(\cdot)$ and that $g(t; C)$ is roughly constant over intervals of length Δ . In this case, not only can the load be uniquely reconstructed from its samples but the arrival processes can also be represented in discrete time. From this point on we use a discrete model for arrivals, departure processes and for the load profile, and replace the previous quantities with their discrete-time counterparts:

$$\begin{aligned} a_q(\ell\Delta) &\rightarrow a_q(\ell), & d_q(\ell\Delta) &\rightarrow d_q(\ell), \\ g(\ell\Delta; C_q) &\rightarrow g_q(\ell), & \hat{L}^S(\ell\Delta) &\rightarrow L^S(\ell). \end{aligned} \quad (12)$$

The discrete-time load synthesis formula from the decisions $d_q(\ell)$ and samples of the pulses $g_q(\ell)$ is:

$$\begin{aligned} L^S(\ell) &= \sum_{q=1}^Q \sum_{k=\ell_0}^{\infty} [d_q(k) - d_q(k-1)] g_q(\ell - k) \\ &= \sum_{q=1}^Q J d_q(\ell) \star g_q(\ell), \quad \ell \geq \ell_0, \end{aligned} \quad (13)$$

where J is the first difference operator. Fig. 1 relates the queues states with the synthesis formula (13).

A. The Inconvenience Cost: Average Delay Model

In order to measure the customers rewards, another quantity that requires modeling is the delay experienced by participants in the DDLS program. Scheduling the energy use of appliances is typically accompanied by deadlines preferred by the consumer, similar to most job scheduling algorithms. But, as in many other scheduling problems that handle diverse job profiles, incorporating deadlines will make the algorithm increasingly complex and not scalable. Specifically, with a random and non-stationary task arrival and resource model, incorporating deadlines would make our problem definition not suitable to handle a large scale customer base. Hence, we follow the path of many scalable computing and communication systems existing today that provide best-effort scheduling services without providing guarantees about the timing of resource delivery to an individual transaction (a great example of which would be the Internet, which currently provides one single class of *best-effort* service to voice and video traffic, which are typically severely degraded by delay). Thus, our algorithm will tolerate occasional deadline misses and will provide the maximum average QoS to the entire population and in return, with much lower communication and computational requirements. This is reasonable since we

imagine that DDLS will be voluntary program and only customers with enough degrees of flexibility in their demand will participate in it. Customers with stricter deadlines shall resort to conventional service providers or be charged more than those accepting best-effort services.

It is a well known result in traffic flow theory that the total delay experienced by the customers in a queue, i.e. $\sum_i (t_i^d - t_i^a)$, is equal to the area of the queue polygon, that represents the state of the queue $s(t)$ versus time, and is obtained by superimposing the departure and arrival profiles, i.e.,

$$s(t) = a(t) - d(t). \quad (14)$$

The delay experienced in the past by the D-loads currently present in the system is not amendable and so, it is not useful in the formulation of the optimization. Hence, for the purpose of optimal scheduling in the future, one can replace the total delay cost with the total delay experienced by all the customers from now on, which we call the *Delay Cost Increment* (DCI):

$$\text{DCI}(t) \triangleq C_I \int_t^{\infty} s(\tau) d\tau = C_I \int_t^{\infty} (a(\tau) - d(\tau)) d\tau, \quad (15)$$

where C_I is the cost per unit of time delay².

With quantization, it is also possible to define different delay costs $C_{I,q}$ for each service queue, allowing to offer higher QoS to customers in return for higher subscription rates. To do so, we define a matrix $C_T = \text{diag}[C_{I,1}, \dots, C_{I,Q}]$:

$$\text{DCI}(t) = \int_t^{\infty} \| C_T(\bar{a}(\tau) - \bar{d}(\tau)) \|_1 d\tau. \quad (16)$$

In addition, when using a discrete set of decision epochs,

$$\text{DCI}(\ell_0) = \left\| \sum_{\ell=\ell_0}^{\infty} C_T[\bar{a}(\ell) - \bar{d}(\ell)] \right\|_1, \quad (17)$$

Where $\ell_0\Delta = \lceil \frac{t}{\Delta} \rceil \Delta$. Note that the interchangeability of integration and the \mathcal{L}_1 norm is possible since $\bar{a}(\tau) - \bar{d}(\tau) \geq 0$.

B. Network model

The communication network supports two directions of communication: the uplink towards the scheduler, and the downlink activation feedback from the scheduler to the loads. We assume that the scheduler aggregates arrivals over a large number of customer premises, each of which pools the requests in a local Home Energy Management System (HEMS), interacting with the scheduler, which we refer as the Community Energy Management System (CEMS). While the CEMS gathers information and schedules the D-loads, another entity provides the interface to the market of generation assets. We call this agent, the Green Energy Management System (GEMS). The architecture is shown in Fig. 2.

The encoding component we laid out above provides a clear basis to compute exactly the rate requirements associated with the DDLS, and in this section we measure the communication rate requirements necessary to enable the control mechanism.

²For a time-dependent delay cost $\text{DCI}(t) = \int_t^{\infty} C_I(\tau) s(\tau) d\tau$.

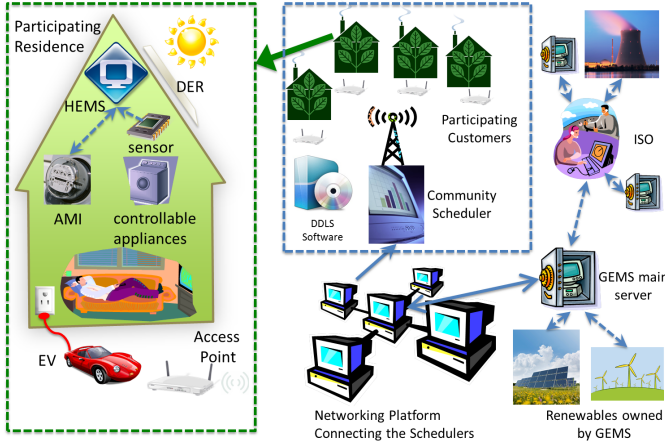


Fig. 2. The GEMS Architecture

1) *The Scheduler Uplink (HEMS to CEMS)*: To reconstruct and optimize the demand, the CEMS needs to know the increments of $\bar{a}(\ell\Delta)$, i.e., the arrivals in the queues.

We envision the HEMS as a software application run by non-dedicated hardware (a personal computer, or a smartphone) and assume that the individual D-loads have a sensor network interface that allows them to convey their request to the HEMS in a local area network (via Bluetooth, Zigbee or new emerging standards for sensor network communications [21]). HEMS locally compute the accrued inconvenience cost, and total consumption of D-loads in the premise over long time intervals. This information is communicated offline to track the quality of service delivered and for billing purposes. However, given that the scheduler only cares about $\bar{a}(\ell\Delta)$, the data communicated for real time scheduling can be both anonymized, as well as aggregated as they flow towards the CEMS. The time sensitive bits correspond to the digital communication of the tuple (t_i^a, q) . Clearly, this requires $\log Q$ bits for the charging class. The arrival time $t_i^a \approx \ell_i^a \Delta$, where ℓ_i^a represents its finite precision representation. Let $\ell_i^n > \ell_i^a$ be the *notification* time index of the arrival, i.e., the time when the event is first recorded and added to the corresponding queue. Let D be the maximum network delay in units Δ , such that with overwhelming probability, $\ell_i^n - \ell_i^a < D$; then, denoting by $\lfloor x \rfloor$ the floor function, this implies that $\lfloor \ell_i^n / D \rfloor = \lfloor \ell_i^a / D \rfloor$. Thus, the code

$$p_i^a = \ell_i^a - \lfloor \ell_i^a / D \rfloor$$

allows to reconstruct the arrival time as $\ell_i^a = \lfloor \ell_i^n / D \rfloor + p_i^a$. In this case, since $p_i^a \in \{0, \dots, D-1\}$ clearly, encoding t_i^a only requires $\log_2 D$ bits. Hence, considering that $\lambda(\ell\Delta)$ is the traffic of D-loads arriving in the system, the HEMS access channel needs to support an aggregate traffic of

$$R_{\text{HEMS}}(\ell) = \frac{1}{\Delta} \lambda(\ell\Delta) \log_2(DQ). \quad (18)$$

As we discussed before, the traffic can be aggregated at the first network relay, acting as a base station (BS); for example a BS could map one to one with each area transformer or to a ISP node, by coalescing the arrival times into information about $\bar{a}(\ell\Delta)$. If the arrivals follow a Poisson arrival process,

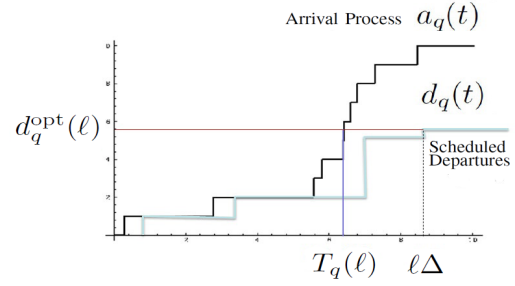


Fig. 3. Mapping decisions into feedback messages

$E\{\bar{a}(\ell\Delta)\} = \lambda(\ell\Delta)$ and the communication rate of the aggregate arrival vector is bounded by:

$$R_{\text{CEMS}}(\ell) = \frac{1}{2} Q \log(2\pi e \lambda(\ell\Delta)). \quad (19)$$

2) *Communicating the decisions back*: Consistent with our uplink model, we envision a downlink message structure that preserves the anonymity of the scheduled user. Once the CEMS decides the optimum schedule, it sends a feedback record, to let the vector $\bar{d}^{\text{opt}}(\ell\Delta)$ of D-loads in. This feedback consists of a $Q \times 1$ vector $\bar{T}(\ell)$, which alerts all appliances in the corresponding classes $q = 1, \dots, Q$ that arrived before times $T_q(\ell)$ to enter the system. The calculation of these vectors is performed as indicated in Fig. 3, i.e.,

$$T_q(\ell) = \max\{\tau \leq \ell : a_q(\tau) \leq d_q^{\text{opt}}(\ell)\}. \quad (20)$$

This system makes sure that the departures match the desired value, while guaranteeing the anonymity of the information about the access. Also in this case it is not necessary to transmit absolute times $T_q(\ell)$. Considering that the delay is an explicit cost for the optimization, and that the optimum decisions are correlated since the queue states are correlated, $T_q(\ell)$ can be differentially encoded with a relatively modest rate requirement. More specifically, assuming that ρ_q is the minimum correlation coefficient among the decisions $d_q(\ell)$ for D-loads in any class and at any time, and that the variance of the delay for appliances is bounded by $\sigma_{T_q}^2$, then each feedback vector requires a number of bits per second:

$$R_{\text{feedback}} = \mathcal{O} \left(\frac{1}{\Delta} \sum_{q=1}^Q \frac{1}{2} \log_2(e(1 - \rho_q^2) \sigma_{T_q}^2) \right).$$

C. Optimal Quantization Rate

Suppose that $\gamma(g(t; C), g(t; Q(C)))$ is a distortion metric, measuring the quantization error in the reproduction of the individual load profile. The average per queue and per load distortion χ_q caused by the load reconstruction technique is,

$$\chi_q = \int_{t=0}^{\infty} \int_{x \in Q^{-1}(C_q)} \gamma(g(t; x), g(t; C_q)) f_C(x) dt dx.$$

Weighting the relative contributions of each queue by the corresponding arrival rate $\lambda_q^{\max} = \sup_t \lambda_q(t)$, the average distortion per unit time χ_{tot} is bounded by

$$\chi_{\text{tot}} \leq \sum_{q=1}^Q \lambda_q^{\max} \chi_q.$$

Hence, the number of codes/queues Q can be chosen based on an appropriate rate-distortion optimization. For example:

1) Minimize Q (and, thus, the rate) while meeting a desired maximum distortion χ^* in load reconstruction, i.e.,

$$\min Q \quad \text{s.t.} \quad \sum_{q=1}^Q \lambda_q^{\max} \chi_q \leq \chi^*.$$

2) Minimize the distortion, limiting the maximum bit rate,

$$\begin{aligned} \min_C \quad & \sum_{q=1}^Q \lambda_q^{\max} \chi_q \\ \text{s.t.} \quad & \max_{\lambda} R_{\text{HEMS}} \leq R_1^{\max} \quad \max_{\lambda} R_{\text{CEMS}} \leq R_2^{\max} \end{aligned}$$

where the rate functions R_{HEMS} and R_{CEMS} are defined in (18)-(19) respectively. It is clear that the optimum in both cases is reached when the constraints are tight.

Next we sketch a few necessary elements of the market interface of our aggregator. We leave the complete development and analysis of this market interface to future work.

IV. THE ELECTRICITY MARKET INTERFACE

We assume that the GEMS cannot use its local renewables to export power to the upstream distribution grid and its generated electricity can only be used locally. The overall system including the customers, renewables, CEMS and GEMS, needs to behave as a single retailer in the wholesale power market and it can also provide ancillary services. The various operational costs that the cell may have include:

1) **Wholesale market day-ahead bidding cost:** by looking on the day-ahead forecasts of its local generation units and its demand pattern, the market interface issues a bid in the day-ahead wholesale market to purchase a certain amount of power for every hour of the next day, so that it can safely serve all of its load reliably.

2) **Wholesale market real-time bidding cost:** if local generation and day-ahead bid are insufficient to meet the demand, the market interface requests more energy from the central grid in the spot market. Also, if it has purchased extra amount of energy that it cannot consume, it can either sell it back to the grid or pay for negative spinning reserve, in case there is no demand for it. To model this cost, we assume that there is what we call the *supply curve* $B(t) + R(t)$, which includes the day-ahead bid plus the the cell local renewable generation and that there is a per unit cost of $C_{up}(t)$ for deviating upward and a per unit cost of $C_{dn}(t)$ for a downward deviation with respect to the supply curve. Subtracting the uncontrollable load $L^N(t)$ from the supply curve, we denote the Zero Incremental Cost (ZIC) power profile for D-loads by $P(t)$:

$$P(t) = B(t) + R(t) - L^N(t). \quad (\text{ZIC - profile}) \quad (21)$$

For short look-ahead horizons, we assume that a reasonably accurate estimate of $R(t)$ is available and thus, $P(t)$ can be considered a deterministic function. Also, we assume that near perfect estimates of the upward and downward balancing prices are available. For results on market clearing price forecasting techniques see [22].

3) **Inconvenience cost paid to customers:** This cost is paid either directly or indirectly to the customer, as an incentive to participate in DDLS programs. We will model this cost using

the concept of DCI introduced in III-A. The inconvenience cost per unit time can be variable with time of day and the queue in which the D-load is placed.

The cell should also purchase energy from its local renewable resources (if they are not owned by the cell). To cover the above mentioned costs, the cell bills its customers for the electricity it purchases from the wholesale market and on-sells to them. Also, a retailer with a DDLS program is in fact capable of providing several types of ancillary services with different dynamics and can provide peak shaving and price control services to the utility or the ISO in return for appropriate reimbursements to cover its costs. We leave a detailed discussion of these issues to our subsequent papers and instead, we will mainly focus on the scenario in which the cell has already bid in the day-ahead market $B(t)$ and has access to the information to determine the ZIC profile in (21), and wants to minimize its costs in real-time.

V. DDLS: THE REAL-TIME SCHEDULING PROBLEM

The GEMS, which is in charge of the market interface of the cell, needs to purchase energy from the central wholesale market in order to satisfy the cell demand. Thus, each day, the GEMS solves an optimization problem to determine the optimal day-ahead bid $B^*(\ell)$, $\ell = 1, \dots, 24$ to be placed in the wholesale market for each hour of the next day. For the rest of this paper, we will assume that the day-ahead bidding process is done and focus on the real-time operation of the cell scheduler. The discussion on how to place these bids based on predictions of the wholesale market price, local generation resources and taking the DDLS ability into account requires considerable amount of space and will be presented in future works (for preliminary results, see [23]).

In real-time, the goal is to minimize the accumulated cost of operation over time. This can be accomplished through a sequential decision maker. In order to derive scheduling decisions that are foresighted, we choose to use a Model Predictive Control (MPC) model. Following our previous notation, $P(\ell)$, $R(\ell)$, $L^S(\ell)$, $C_{up}(\ell)$ and $C_{dn}(\ell)$ are samples of their continuous counterparts at $t = \ell\Delta$ (the costs are also multiplied by the interval length Δ). Assuming a look-ahead horizon of T units of time, and denoting by u_q the duration of the jobs of appliances in the q -th queue, we can formulate the general real-time scheduling problem for the cell as,

$$\begin{aligned} D^* = \operatorname{argmin}_D \quad & \mathbb{E}_A[\text{Cost of retail entity in real time}] \\ & \sum_{\ell=\ell_0}^{\ell_0+T} \mathbb{E}_A\{C_\ell(L^N(\ell) + L^S(\ell, D), B^*(\ell), R(\ell)) \\ & \quad + \text{DCI}(\ell_0, D)\} \end{aligned} \quad (22)$$

$$\begin{aligned} \text{s.t.} \quad & \forall \ell \leq \ell_0 + T, q \in \{1, \dots, Q\}: d_q(\ell-1) \leq d_q(\ell) \leq a_q(\ell) \\ & d_q(\ell) \in \mathbb{N}, \quad d_q(\ell + T - u_q) = a_q(\ell + T - u_q), \end{aligned}$$

where $C_\ell(\cdot)$ represents a prediction of the *real-time* cost that the cell incurs at epoch ℓ when purchasing electricity equal to $L^N(\ell) + L^S(D, \ell) - (B^*(\ell) + R(\ell))$ from the local intermittent resources or a central wholesale market (if negative, the retail utility may have to pay for negative spinning reserve).

The matrix D represents the set of scheduling decisions that will be made in the look-ahead horizon, i.e. $D = [\bar{d}(\ell), \dots, \bar{d}(\ell_0 + T)]$ and D^* represents its optimal value. The matrix $A = [\bar{a}(\ell), \dots, \bar{a}(\ell_0 + T)]$ represents the arrival process. \mathcal{D} is the entire decision space, i.e., $D \in \mathcal{D}$ and \mathcal{A} is the space defined by the arrival process, i.e., $A \in \mathcal{A}$. $\text{DCI}(\ell_0, D)$ is the cumulative cost of delaying the customers, calculated in a finite horizon T . The first and second constraints are due to causality and the third constraint requires that all the arriving appliances finish their jobs by time $\ell_0 + T$. Stricter deadlines can be imposed to provide higher QoS to certain queues.

Considering the costs defined in the Section IV, and denoting respectively by $P_{up}(\ell)$ and $P_{dn}(\ell)$ the upward and downward deviations from the ZIC power supply $P(\ell)$ in (21) for the D-loads, the scheduling problem in (22) can be formulated as the following mixed integer optimization problem,

$$\begin{aligned} \min_D \quad & \mathbb{E} \left\{ \sum_{\ell=\ell_0}^{\ell_0+T} [C_{up}(\ell)P_{up}(\ell) + C_{dn}(\ell)P_{dn}(\ell)] \right. \\ & \left. + \sum_{q=1}^Q C_{I,q}(a_q(\ell) - d_q(\ell)) \right\} \end{aligned} \quad (23)$$

$$\text{s.t. } \forall \ell \leq \ell_0 + T, q \in \{1, \dots, Q\}: d_q(\ell-1) \leq d_q(\ell) \leq a_q(\ell)$$

$$d_q(\ell) \in \mathbb{N}, \quad d_q(\ell + T - u_q) = a_q(\ell + T - u_q),$$

$$\forall \ell \leq \ell_0 + T: P_{up}(\ell) \geq 0, \quad P_{dn}(\ell) \geq 0, \quad P_{dn}(\ell)P_{up}(\ell) = 0,$$

$$P_{up}(\ell) - P_{dn}(\ell) + P(\ell) = \Re\{L^S(\ell, D)\},$$

where $\Re[L^S(\ell, D)]$ represents the active (real) part of the D-load for $\ell > \ell_0$.

After solving the above optimization problem at time ℓ_0 , the scheduler informs the D-loads about its decisions on the optimum departure process $d_q(\ell_0)$ for each queue, and discards the decisions made for the future (the dummy variables). Afterwards, it needs to solve the optimization at time $\ell_0 + 1$ again. Since we assume that scheduled tasks are not interruptible, the load due to D-loads that were scheduled to start their job at time ℓ_0 is added to the non-controllable part of the load at and after time $\ell_0 + 1$, i.e., for $\ell \geq \ell_0 + 1$:

$$L^N(\ell) \rightarrow L^N(\ell) + \sum_{q=1}^Q [d_q(\ell_0) - d_q(\ell_0 - 1)]g_q(\ell - \ell_0).$$

Remark 5.1: The finite horizon assumption is made since $P(\ell)$ is non-stationary and our knowledge about it cannot be considered perfect in an infinite horizon. Also, forecast errors for the arrival pattern of customers increases with time.

Remark 5.2: In the following, we will ignore the reactive power load due to the scheduled D-loads and assume that the pulses $g_q(\ell)$ will represent active power requests only (i.e., $\Re[L^S(\ell)] = L^S(\ell)$). But, if desired, one can easily incorporate reactive power requirements in the above optimization problem as a constraint:

$$\Im[L^S(\ell)] = \sum_{q=1}^Q \sum_{i=\ell_0}^{\ell} (d_q(i) - d_q(i-1))\Im[g_q(\ell-i)] \leq \epsilon.$$

For brevity, from this point on we omit the dependency on D of both $L^S(\ell; D)$ and $\text{DCI}(\ell_0, D)$. Next, we relax the problem (23), considering two scenarios for the arrival process.

A. Scenario I

In this scenario, we look at the case where the arrival process is deterministic but the rate $\lambda_q(t)$ is not constant in time. This problem can be formulated as (23) without the expected value operator.

Lemma 5.3: If we ignore the case where $C_{dn}(\ell) < 0$, the constraint $P_{dn}(\ell)P_{up}(\ell) = 0$ can be eliminated from (23).

Proof: (by contradiction). Assume that for epoch ℓ , the optimum solution to (23) is such that $P_{dn}^*(\ell)P_{up}^*(\ell) > 0$. One can replace these values by setting the smaller number to 0 and the larger one to $|P_{dn}^*(\ell) - P_{up}^*(\ell)|$. The constraints will all still hold while the cost will be reduced. Thus, the optimization problem without the constraint $P_{dn}^*(\ell)P_{up}^*(\ell) = 0$ is equivalent to (23). ■

To continue, we define the following matrices,

$$\begin{aligned} D &= [\bar{d}(\ell_0), \dots, \bar{d}(\ell_0 + T)] \\ A &= [\bar{a}(\ell_0), \dots, \bar{a}(\ell_0 + T)] \\ P &= [P(\ell_0), \dots, P(\ell_0 + T)] \\ P_{up} &= [P_{up}(\ell_0), \dots, P_{up}(\ell_0 + T)] \\ P_{dn} &= [P_{dn}(\ell_0), \dots, P_{dn}(\ell_0 + T)] \\ C_{up} &= [C_{up}(\ell_0), \dots, C_{up}(\ell_0 + T)] \\ C_{dn} &= [C_{dn}(\ell_0), \dots, C_{dn}(\ell_0 + T)] \\ C_T &= [C_{I,1}, \dots, C_{I,Q}]. \end{aligned} \quad (24)$$

Lemma 5.4: (23) can be cast in the following constrained mixed-integer linear programming problem:

$$\begin{aligned} \min_{D, P_{up}, P_{dn}} \quad & C_{up}P_{up}^T + C_{dn}P_{dn}^T - (C_T \otimes \mathbf{1}_{1 \times L})\text{vec}(D^T) \quad (25) \\ \text{s.t.} \quad & 0 \preceq \text{vec}(D^T) \preceq \text{vec}(A^T), \quad (I \otimes J^T)\text{vec}(D^T) \succeq 0, \\ & P_{up} \succeq 0, \quad P_{dn} \succeq 0, \quad \Gamma \text{vec}(D^T) + P^T + P_{up}^T - P_{dn}^T = 0, \\ & D \in \mathbb{N}^{Q \times (T+1)} \end{aligned}$$

where vec denotes the vectorization operation. The matrix $\Gamma = [\Gamma_1, \Gamma_2, \dots, \Gamma_Q](I \otimes J^T)$, where J is the first difference operator, and the matrices Γ_q are the Toeplitz matrices associated with each $g_q(\ell) = g(\ell; C_q)$, with first row equal to $[g_q(1), 0, \dots, 0]$ and first column equal to $[g_q(1), g_q(2), \dots, g_q(T)]$.

Proof: We showed that the load due to D-loads is the sum of the following Q functions:

$$\forall q: L_q^S(\ell) = \sum_{i=\ell_0}^{\ell_0+T} (D(q, i) - D(q, i-1))g_q(\ell-i) \quad (26)$$

which is basically the convolution of first difference of the q^{th} row of D with the vector $[g_q(1), \dots, g_q(T)]$. Thus, using Q Toeplitz matrices, each associated with one of the $g_q(\ell)$'s, and considering that $\text{vec}((DJ)^T) = (I \otimes J^T)\text{vec}(D^T)$, we can rewrite the load due to D-loads as the following vector

$$L^S = [\Gamma_1, \Gamma_2, \dots, \Gamma_Q](I \otimes J^T)\text{vec}(D^T).$$

On the other hand, the DCI in (16) can be rewritten as

$$\|C_T(A - D)\|_1 = (C_T \otimes \mathbf{1}_{1 \times L})(\text{vec}(A^T) - \text{vec}(D^T)). \quad (27)$$

With the first term independent of the decision variables and thus, not contributing to the objective function minimization, we can add only the second term to the cost to represent the delay cost. The rest of the constraints in (25) are just transformations of the constraints in (23) to vector form. ■

Since large-scale integer programs like (25) are complex to solve, we may choose to relax the problem by omitting the integrality constraints for the decision variables. The linear program obtained gives a lower bound on the optimal value of the integer linear programming problem (the feasible region of the integer program is a subset of its corresponding LP relaxation problem). After solving the relaxed problem, we can round the fractional values to obtain integral solutions. A possible modification is to try relaxations based branch and bound algorithms and recurse into several relaxed subproblems until a good solution is found. Due to the large size of our problem, we resort to the simpler rounding solution.

B. Scenario II

In this scenario, customers arrive according to a random arrival process with a non-homogeneous rate $\lambda(t)$, e.g. a Poisson process. Since we no longer have deterministic information about future arrivals and only the past is fully observable for us, we need to solve a stochastic shortest path problem:

$$C_{\ell_0}^*(\bar{x}(\ell_0)) = \min_{\bar{a}} \sum_{\ell=\ell_0}^{\infty} E_{\bar{a}} \{c(\bar{x}(\ell), \bar{d}(\ell), \bar{a}(\ell))\}, \quad (28)$$

where the vector $\bar{x}(\ell)$, representing the state (expanded to include previous decisions), and the function $c(\cdot)$ representing the cost incurred at time ℓ are defined as

$$\bar{x}(\ell) = [P(\ell), \bar{d}(\ell - \min\{\ell - \ell_0, S\}), \dots, \bar{d}(\ell - 1)]^T,$$

$$\begin{aligned} c(\bar{x}(\ell), \bar{d}(\ell), \bar{a}(\ell)) &= C_{up}(\ell)P_{up}(\ell) + C_{dn}(\ell)P_{dn}(\ell) \\ &+ \sum_{q=1}^Q C_{I,q}(a_q(\ell) - d_q(\ell)). \end{aligned}$$

A state variable is the minimal history that is necessary and sufficient to compute the decision function [24]. For our case, this includes all the decisions made in the past that can affect the current load of the system and the ZIC power profile in (21). If we assume that the appliances can be on for a maximum duration of $S\Delta$ units of time, the last S decisions should be included in the state variable.

As our knowledge about the intermittent resources and the arrival trends of D-loads will degrade significantly with time, solving (28) is not practical. Based on this fact, we divide our look-ahead horizon into the categories below:

- 1) Full statistical knowledge interval: in this interval, we have full statistical information about the arrival process and the amount of available resource and thus, we exploit this knowledge in our decision strategy.
- 2) Imperfect information interval: during this interval, we have some information about the arrival and generation processes

but our knowledge is not as accurate as the first interval and thus, we will only use the expected value of predictions.

- 3) No useful knowledge interval: after a certain amount of time, our knowledge about the arrival process and the resource will degrade so significantly that we would rather not incorporate them in our decision.

Hence, we redefine our optimized scheduling problem as:

$$\min_{D \in \mathcal{D}} E_{\bar{x}} \left\{ \sum_{\ell=\ell_0}^{\ell_0+T-1} c(\bar{x}(\ell), \bar{d}(\ell), \bar{a}(\ell)) + \hat{C}_{\ell_0+T}(\bar{x}(\ell_0+T)) \right\} \quad (29)$$

where $\hat{C}_{\ell_0+T}(x(\ell_0+T))$ is the approximation of the true cost-to-go function $C_{\ell_0+T}^*(x(\ell_0+T))$ after T stages (the length of the full statistical knowledge interval). This approximation is found using a base suboptimal policy that we solve for assuming that the uncertain quantities in the imperfect information interval are fixed at their typical values. Thus, this is an T -step lookahead policy, with the optimal cost-to-go approximated by the cost-to-go of a certainty equivalent controller. Note that we have already discussed the possible strategies to solve this problem in scenario III. Also, we can reduce the number of possible states by using state aggregation strategies in the complete statistical knowledge interval in order to construct a more simple and tractable problem. This will make the computation much less demanding.

Remark 5.5: We solved the scheduling problem by ignoring flow constraints of the distribution network. Since we do not take individual D-load delays into account and we only care about the total delay experienced by the customers, these flow constraints can be incorporated in the design with a simple modification using the following suboptimal policy: If the BS knows that it does not have distribution capacity to let any more appliances start functioning, it sends a full capacity message to the CEMS, which can be incorporated into (20).

VI. NUMERICAL RESULTS

In this section, we compare the effectiveness of the introduced scheduling policies exercised at a cell that controls the charging of a fleet of around 15,000 vehicles to the following two scenarios: the case where no control mechanism is provided over the load and the case where the control is applied in a more distributed fashion with 500 local schedulers spread across the area where the program is implemented. We assumed that the following information are available to the scheduler: 1) perfect predictions of the mean arrival rate of vehicles $\lambda(t)$; 2) causal observability of the arrival process $a(t)$; 3) predictions of the ZIC power profile (21) in the lookahead horizon; 4) predictions of the upward and downward clearing prices in the lookahead horizon; and 5) accurate estimates of the load injections $g_q(t)$. To avoid computational complexities and to make the resulting demand profiles more meaningful to the reader, we assumed that the charging duration of the vehicles has a maximum of 8 hours and is quantized in 15 minute intervals, which is equal to the frequency with which the scheduler solves the optimization problem. The appliances arrive in the system following a Poisson arrival process with a mean arrival rate $\lambda = 12$ per hour for each of the 32 queues, assumed to be constant, in order for the

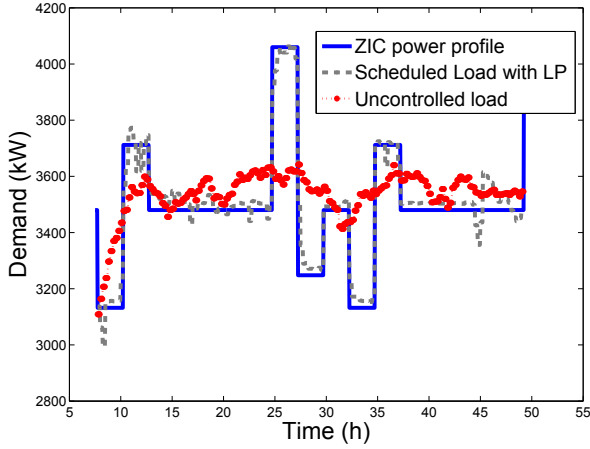


Fig. 4. DDLS allows to follow a certain generation profile

results to be interpretable. The optimizers solve the scheduling problem with a certainty equivalent controller that uses a relaxed linear programming approach to determine the best scheduling strategy. After each step, the scheduled D-loads are added to the uncontrollable load term for future epochs. The upward and downward balancing prices were chosen to be equal and the charging pulses were assumed to be square pulses scaled to match the duration of charging.

Fig. 4 compares the demand in the following two scenarios: coordinated charging of all vehicles by one scheduler unit and uncontrolled charging. The look-ahead horizon in these simulations is assumed to be 8 hours and to be fair, no appliance is allowed to be delayed beyond 8 hours after it's arrival, which in reality can differ for different queues. Thus, the number of appliances receiving service is equal in the two scenarios. The results show that the average delay experienced by customers if everybody participates in the DDLS program is less than one unit of time and due to the considerable decrease in the deviation costs, there is a relative 41 percent reduction in operational costs compared to the uncontrolled case.

Fig. 5 compares the case where instead of a single scheduler unit, several schedulers are distributed across the area where the DDLS program is carried out (20 schedulers in our case, each scheduling around 20 EVs per hour). To showcase the more subtle differences in this comparison, we have scaled down our simulation horizon to 16 hours and decreased the arrival rate by half. The savings that resulted from the distributed schedulers were still considerable (21 percent).

In Fig. 6, we compare the load directly controlled by the DDLS scheduler to the case where a single price signal is broadcast to every vehicle, using which they locally minimize their cost for charging. Due to a lack of work that examines how to optimally design the pricing signal and, since the comparison is based on how well the controlled load can follow the day-ahead bid, the price signals were chosen to be a function of excess or shortfall of generation from the uncontrolled load profile (which is close to flat). Individual customers are provided with the price for the whole day. Several price functions were tested but they all led to similar

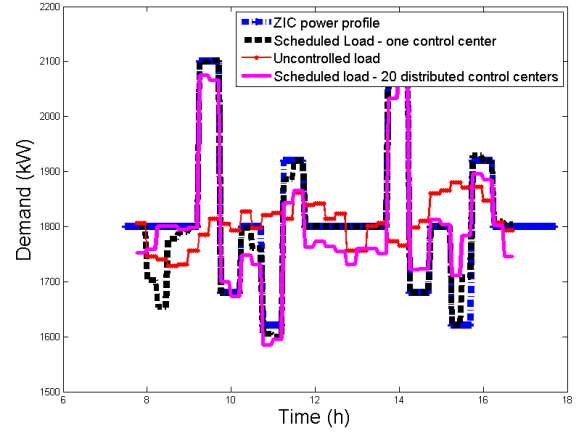


Fig. 5. Higher level of aggregation leads to greater savings

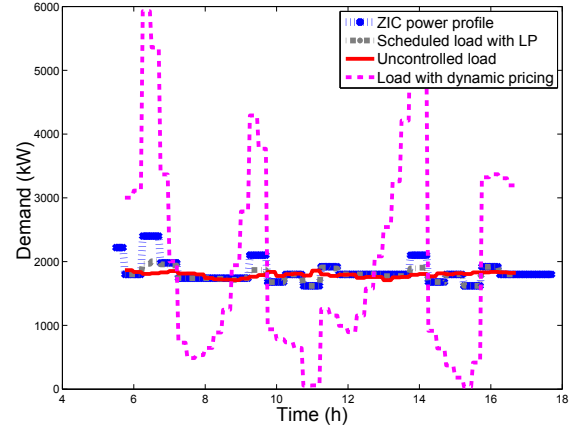


Fig. 6. A single pricing signal for everybody results in rebound peaks

results. Since customers act selfishly to minimize their own costs, this technique results in very high rebound peaks, occurring at times when the price is lower even by a small amount (due to small increases in the ZIC power profile in (21)). The conclusion that can be drawn from this result is that implementing pricing techniques, irrespective of how good the individual HEMS work, needs much more research. Also, sending a single price signal to all the customers can endanger the stability of the grid.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced a novel DSM model for direct load scheduling. We provided a communication framework to defer the time at which loads turn on, after informing the control center about their arrival in the grid and service request. We described a cellular architecture that we consider suitable to integrate Digital Direct Load Scheduling programs into the grid. Finally, we formulated and solved the problem of managing the energy demand from loads that can be deferred during the real-time operation of the cell. Simulations showed that such a program can alleviate the problem of matching demand and generation considerably with minimal inconvenience to the customers and parsimonious communication

needs. In our future work, we will study architectures to incorporate these large reservoirs of storable demand in the electricity market. We will also look more closely at the statistical information that is required to solve the scheduling problems, e.g. the arrival patterns of D-loads, their requests and the availability of renewables. Another major area for future work would be to look at how to integrate the degrees of freedom offered by thermostatically controlled appliances in a neighborhood DSM program (maybe with smaller coverage areas than what was considered in this paper).

REFERENCES

- [1] D. E. King, "Electric power micro-grids: Opportunities and challenges for an emerging distributed energy architecture," *PhD Thesis, Carnegie Mellon University*, 2006.
- [2] R. Brown, "Impact of smart grid on distribution system design," in *Power and Energy Society General Meeting 2008*, July 2008, pp. 1–4.
- [3] S. Zeng, J. Li, and Y. Ren, "Research of time-of-use electricity pricing models in china: A survey," in *IEEM 2008*, Dec 2008, pp. 2191–2195.
- [4] Y. Jia-hai, "Customer response under time-of-use electricity pricing policy based on multi-agent system simulation," in *Power Systems Conference and Exposition, IEEE PES*, nov 2006, pp. 814–818.
- [5] A.-H. Mohsenian-Rad and A. Leon-Garcia, "Optimal residential load control with price prediction in real-time electricity pricing environments," *Smart Grid, IEEE Trans. on*, vol. 1, no. 2, pp. 120–133.
- [6] D.-M. Han and J.-H. Lim, "Design and implementation of smart home energy management systems based on zigbee," *Consumer Electronics, IEEE Transactions on*, vol. 56, no. 3, pp. 1417–1425, 2010.
- [7] S. Kishore and L. Snyder, "Control mechanisms for residential electricity demand in smartgrids," in *Smartgridcomm 2010*, pp. 443–448.
- [8] M. Roozbehani, M. A. Dahleh, and S. K. Mitter, "Volatility of power grids under real-time pricing," arXiv.org, Tech. Rep., June 2011.
- [9] K. Y. Huang, "Demand subscription services-an iterative dynamic programming for the substation suffering from capacity shortage," *Power Systems, IEEE Transactions on*, vol. 18, no. 2, pp. 947–953, May 2003.
- [10] M. Pedrasa, T. Spooner, and I. MacGill, "Scheduling of demand side resources using binary particle swarm optimization," *Power Systems, IEEE Transactions on*, vol. 24, no. 3, pp. 1173–1181, Aug 2009.
- [11] K. Valentine, W. Temple, and M. Zhang, "Intelligent electric vehicle charging: Rethinking the valley-fill," *Journal of Power Sources*, vol. 196, no. 24, pp. 10717–10726, 2011.
- [12] T. Kim and H. Poor, "Scheduling power consumption with price uncertainty," *Smart Grid, IEEE Trans. on*, vol. 2, no. 3, pp. 519–527, Sep. 2011.
- [13] J. Xiao, J. Y. Chung, J. Li, R. Boutaba, and J.-K. Hong, "Near optimal demand-side energy management under real-time demand-response pricing," in *CNSM 2010*, Oct. 2010, pp. 527–532.
- [14] M. C. Caramanis and J. Foster, "Management of electric vehicle charging to mitigate renewable generation intermittency and distribution network congestion," *48th IEEE CDC*, pp. 4717–4722, Dec 2009.
- [15] S. Sojoudi and S. H. Low, "Optimal charging of plug-in hybrid electric vehicles in smart grids," in *IEEE Power and Energy Society General Meeting*, Detroit, MI, July 2011.
- [16] T. Lambert, P. Gilman, and P. Lilienthal, *Micropower System Modeling with Homer*. John Wiley and Sons, Inc., 2006, pp. 379–418.
- [17] A. Papavasiliou and S. Oren, "Coupling wind generators with deferrable loads," in *Energy 2030 Conference, 2008*.
- [18] G. Gross and F. Galiana, "Short-term load forecasting," *Proceedings of the IEEE*, vol. 75, no. 12, pp. 1558–1573, Dec 1987.
- [19] K.-H. Kim, J.-K. Park, K.-J. Hwang, and S.-H. Kim, "Implementation of hybrid short-term load forecasting system using artificial neural networks and fuzzy expert systems," *Power Systems, IEEE Transactions on*, vol. 10, no. 3, pp. 1534–1539, Aug 1995.
- [20] S.-J. Huang and K.-R. Shih, "Short-term load forecasting via arma model identification including non-gaussian process considerations," *Power Systems, IEEE Transactions on*, vol. 18, no. 2, pp. 673–679.
- [21] D. Culler, D. Estrin, and M. Srivastava, "Guest editors' introduction: Overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41–49, Aug. 2004.
- [22] M. Olsson and L. Soder, "Estimating real-time balancing prices in wind power systems," in *PSCE '09*, Mar 2009, pp. 1–9.
- [23] M. Alizadeh, T. Chang, A. Scaglione, C. Chen, and S. Kishore, "The emergence of deferrable energy requests and a greener future: What stands in the way?" in *ISCCSP 2012*.
- [24] W. B. Powell, *Approximate Dynamic Programming I: Modeling*. John Wiley and Sons, Inc., 2010.



Mahnoosh Alizadeh (S'08) has been a PhD student in University of California, Davis since 2009. She received her B.S. degree in Electrical Engineering from Sharif University of Technology. Her research interests mainly lie in the area of Smart Grid, in particular demand side management and demand response.



Anna Scaglione (S'97-M'99-SM'09-F'11) received the Laurea (M.Sc. degree) and the Ph.D. degree from the University of Rome La Sapienza, Italy, in 1995 and 1999, respectively. She was previously at Cornell University from 2001 to 2008, where she became Associate Professor in 2006; before joining Cornell, she was Assistant Professor from 2000 to 2001, at the University of New Mexico. Since July 2008, she has been Associate Professor in Electrical and Computer Engineering at the University of California, Davis. Her expertise is in the broad area

of signal processing for communication systems and networks. Her current research focuses on cooperative wireless networks and sensors systems for monitoring and control applications



Robert J. Thomas (M'73-SM'83-F'93-LF'08) received the Ph.D. degree in electrical engineering from Wayne State University, Detroit, MI, in 1973. He is currently Professor Emeritus of Electrical and Computer Engineering at Cornell University, Ithaca, NY. His technical background is broadly in the areas of systems analysis and control of large-scale electric power systems. He has published in the areas of transient control and voltage collapse problems as well as technical, economic, and institutional impacts of restructuring. Prof. Thomas is a member

of Tau Beta Pi, Eta Kappa Nu, Sigma Xi, and ASEE. He has received five teaching awards and the IEEE Centennial and Millennium medals. He has been a member of the IEEE-USA Energy Policy Committee since 1991 and was the committee's Chair from 1997-1998. He is the founding Director of the 13-university-member National Science Foundation Industry/University Cooperative Research Center, PSERC and he currently serves as one of 30 inaugural members of the U.S. Department of Energy Secretary's Electricity Advisory Committee (EAC).